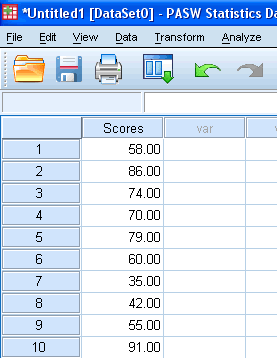
# Recoding Variables in SPSS Statistic| If condition

**Example:** The data given below represents the scores of 10 students in a final examination. Recode the data giving code "1" to scores between 75 - 100, code 2 to scores between 61 - 74, code 3 to scores between 41 - 60 and code 4 to scores between 0 - 40.

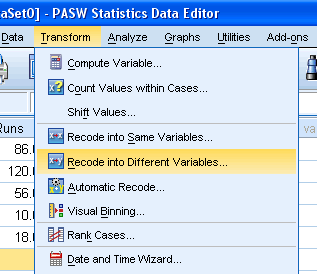
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Final examination scores of 10 students | | | | | | | | | | |
| Scores | 58 | 86 | 74 | 70 | 79 | 60 | 35 | 42 | 55 | 91 |

Enter the data in the SPSS Statistics Data Editor and name the variable "**Scores**".

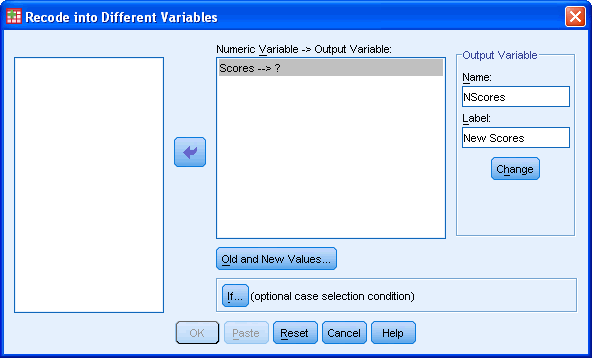
*Remember that each individual's results go on a separate line (row) in SPSS Statistics.*



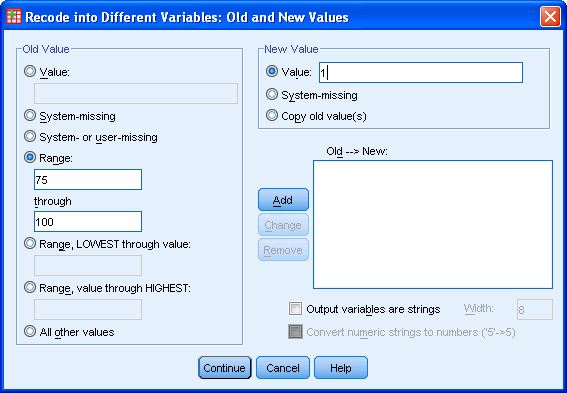
Click on **Transform > Recode Into Different Variables...** in the top menu.



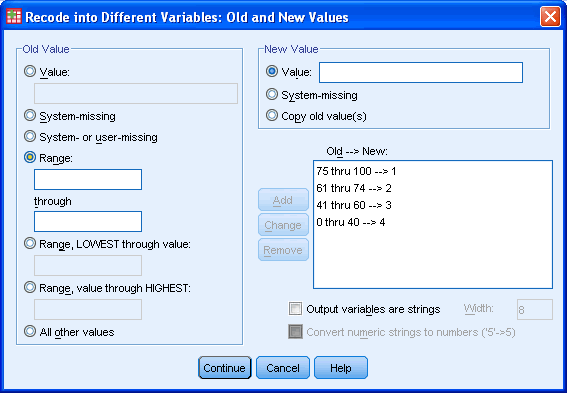
Transfer the variable you want to recode by selected it and pressing the SPSS Right Arrow Button button, and give the new variable a name and label. In this example, we have given the new variable a name of "**NScores**" and label of "**New Scores**" as shown below:



* Click the SPSS Change Button button.
* Click on the SPSS Old and New Values Button button.
* Enter the first range of "**75 - 100**" into the Range: box within the -Old Value- area, and set the new code to "**1**" into the Value: box within the -New Value- area, as shown below:



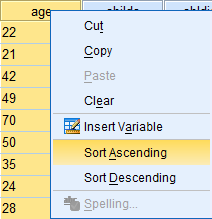
* Click the SPSS Add Button button.
* Repeat for all other values, such that you are presented with the following screen:



**Sort Data**

Columns of data in the Data View of the Data Editor window can be sorted in ascending or descending order.

To sort data, right click on the column heading (the name of the variable you wish to sort) and click on the desired sort order (ascending or descending). The variable will be sorted along with each case's corresponding data values in the other variable categories.

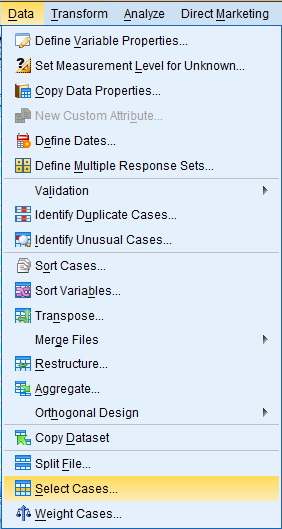


**Select Cases: Filter**

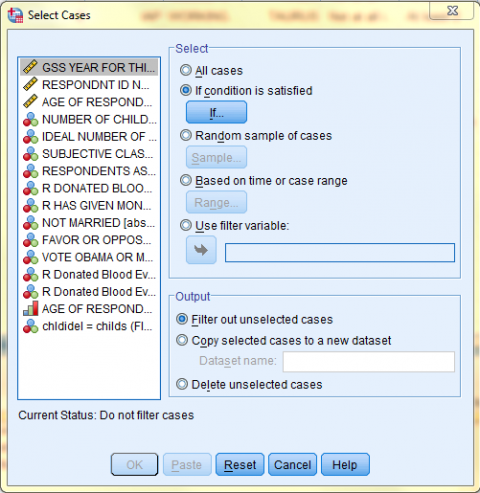
**Filter According to One Condition:**

In this example, we want to select data from respondents who are married and filter out all other data. Specifically, if the variable absingle = 2, then the data from respondents who are married will be selected, because 2 represents respondents who are married. All other cases will be filtered out. When cases are selected and other data is filtered out, all further analyses and graphs reflect only the selected data (until the filter is turned off).

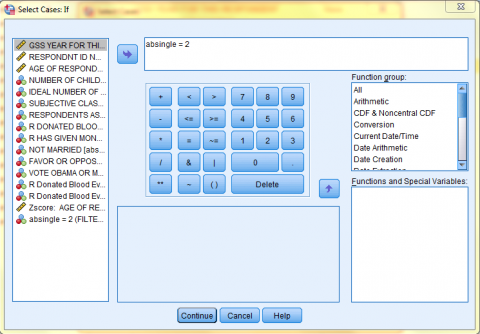
To use the Select Cases function, click on 'Data' in the toolbar at the top of the Data Editor window, and then click on 'Select Cases' in the dropdown menu.



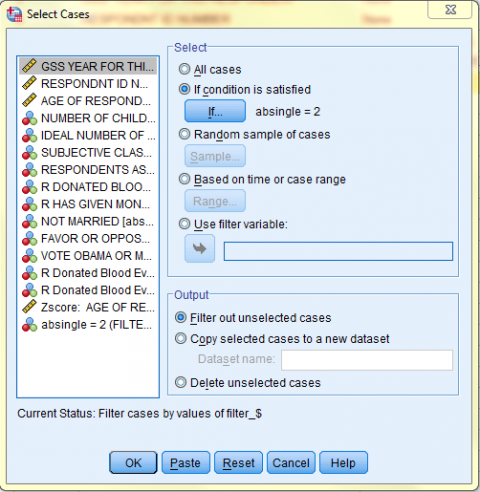
In the dialog box that pops up, the 'Select' field offers a few different methods of filtering data. In the 'All cases' option, all data cases are selected and none are filtered out. In the 'If condition is satisfied' option, data is filtered out in accordance with a predetermined condition. Select the 'If condition is satisfied' option and click 'If...' to specify the filtering conditions. For example, if you are interested in respondents that are married, then you would want to select those cases. In our dataset, marital status is denoted by the 'absingle' variable, and marriage is denoted with a 2. Hence, the specific condition is: absingle = 2.

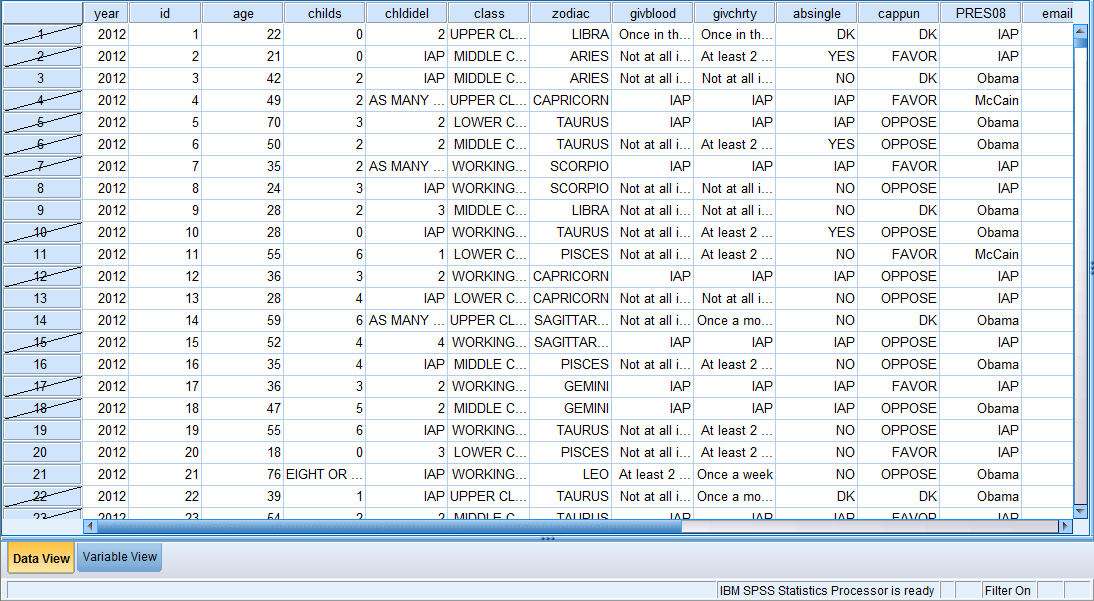


In the Select Cases: If dialog box that pops up, enter the specific conditions according to which you would like data to be selected and filtered. Then, click 'Continue.' In this example, the specific condition is: absingle = 2. All cases where the respondent is married will be selected, and all other cases will be filtered out.



Back in the Select Cases dialog box, the If Condition is now displayed next to the 'If...' button. Next, choose your desired output option and click 'OK.' In this example, we want to 'Filter out unselected cases.'



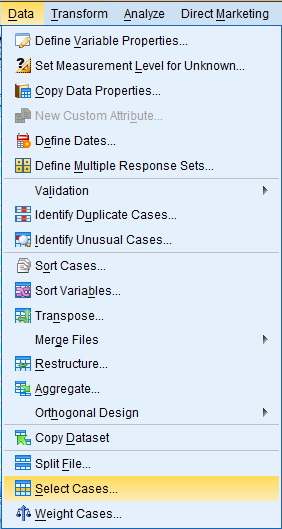
The following image displays the Data View of the Data Editor with the 'absingle = 2' filter on. Note how the bottom right corner of the screen says 'Filter On.' Remember, all further analyses and graphs will reflect only the selected data (until the filter is turned off).

To turn the filter off, click on 'Data' in the toolbar at the top of the Data Editor window, and then click on 'Select Cases' in the dropdown menu. Then, click on 'All cases' to deselect the 'If condition is satisfied' option, and click 'OK'.

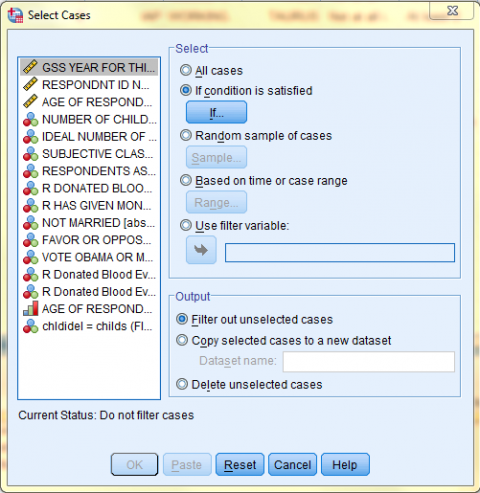
**Filter According to Multiple Conditions:**

In this example, we want to select data from respondents who are in the middle class or upper class and above the age of 50, and we want to filter out all other data. Specifically, if the variable 'class' is equal to 3 or 4, and the variable 'age'  is greater than or equal to 50, then the data will be selected (because the 'class' values 3 and 4 correspond with respondents who are in the middle and upper class, and the 'age' values greater than or equal to 50 correspond with respondents who are 50 years old and above. All other cases will be filtered out. When cases are selected and other data is filtered out, all further analyses and graphs reflect only the selected data (until the filter is turned off).

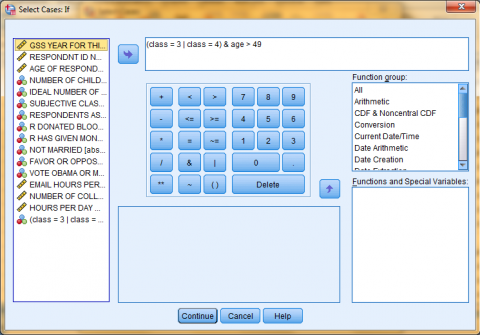
To use the Select Cases function, click on 'Data' in the toolbar at the top of the Data Editor window, and then click on 'Select Cases' in the dropdown menu.



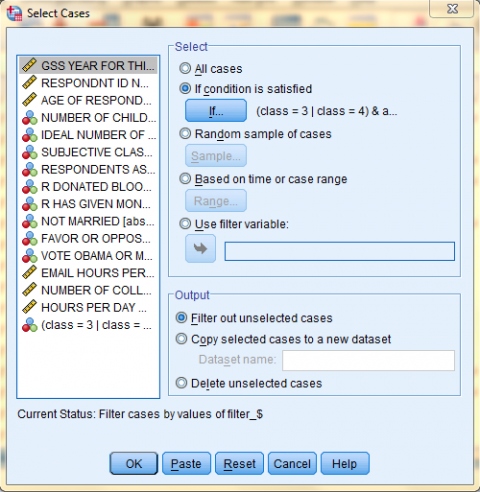
In the dialog box that pops up, the 'Select' field offers a few different methods of filtering data. In the 'All cases' option, all data cases are selected and none are filtered out. In the 'If condition is satisfied' option, data is filtered out in accordance with a predetermined condition. Select the 'If condition is satisfied' option and click 'If...' to specify the filtering conditions. For example, if you are interested in respondents who are in the middle or upper class and are above the age of 49, then you would want to select those cases. In our dataset, class is denoted by the 'class' variable, and the 'class' values 3 and 4 correspond with respondents who are in the middle and upper class. Age is denoted by the 'age' variable, and the 'age' values greater than 49 correspond with respondents who are 50 years old and above. Hence, the specific condition is: (class = 3 | class = 4) & age > 49.

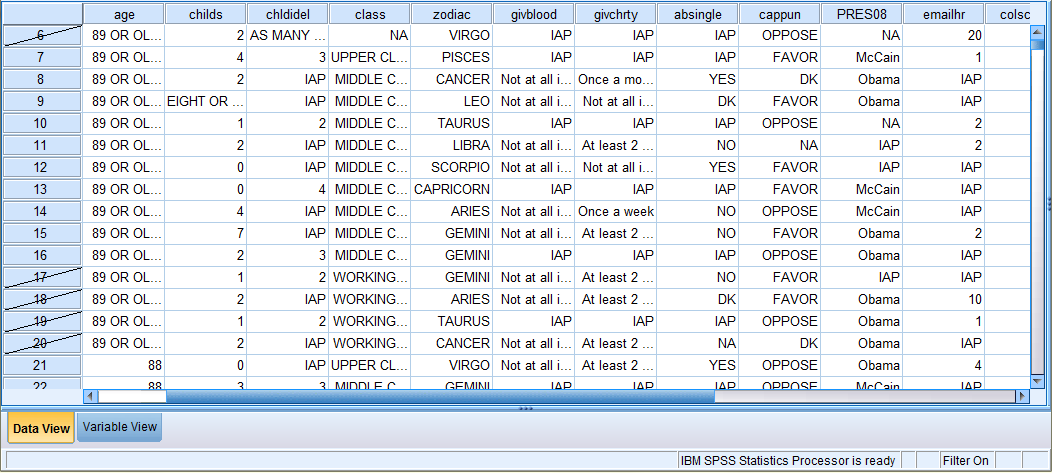


In the Select Cases: If dialog box that pops up, enter the specific conditions according to which you would like data will be selected and filtered. Then, click 'Continue.' In this example, the specific condition is: '(class = 3 | class = 4) & age > 49.' All cases where the respondents are in the middle or upper class and are above the age of 49 will be selected, and all other cases will be filtered out.



Back in the Select Cases dialog box, the If Condition is now displayed next to the 'If...' button. Next, choose your desired output option and click 'OK.' In this example, we want to 'Filter out unselected cases.'



The following image displays the Data View of the Data Editor with the '(class = 3 | class = 4) & age > 49' filter on. Note how the bottom right corner of the screen says 'Filter On.' Remember, all further analyses and graphs will reflect only the selected data (until the filter is turned off).

To turn the filter off, click on 'Data' in the toolbar at the top of the Data Editor window (specifically in the Data View), and then click on 'Select Cases' in the dropdown menu. Then, click on 'All cases' to deselect the 'If condition is satisfied' option, and click 'OK'.

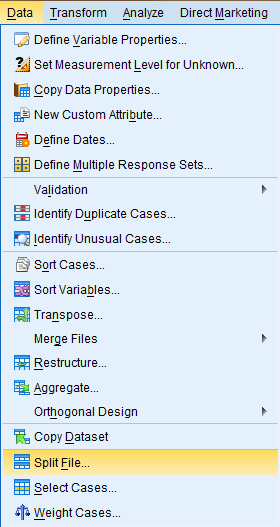
Lastly, SPSS can filter a 'Random sample of cases,' cases 'Based on time or case range,' or based on a filter variable (filter variables are created each time a filter is made using any of the other filter options).

Split File:

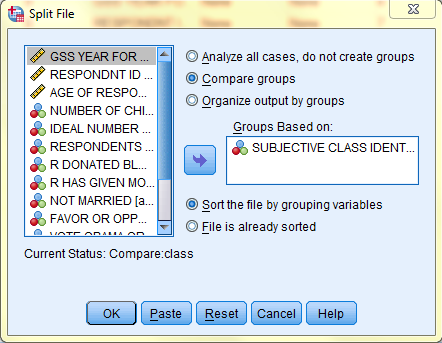
The Split File function in SPSS enables users to sort data and organize output by subgroups. This is useful when comparing data among groups.

In this example, we will split the file (and all output) according to respondents' subjective class identification. Spefically, when a frequency table is generated, six individual frequency tables will be in the output, one for each class group (Lower, Working, Middle, Upper, DK, and NA).

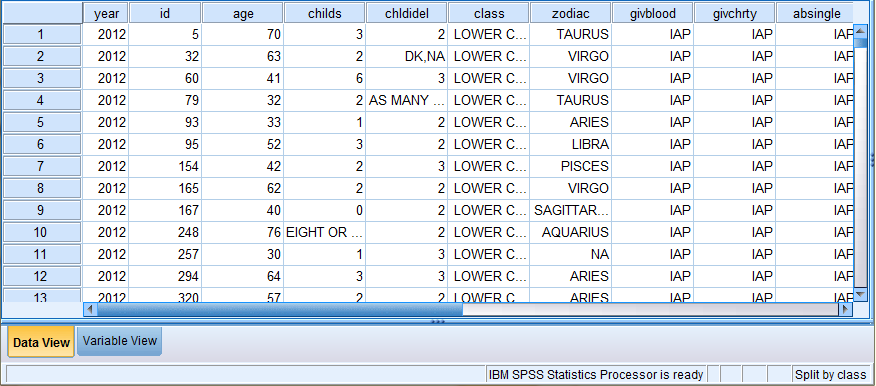
To use the Split File Cases function, click on 'Data' in the toolbar at the top of the Data Editor window, and then click on 'Split File' in the dropdown menu.



In the Split File dialog box that pops up, a few different methods of splitting data are offered. In the 'Analyze all cases, do not create groups' option, all data cases are selected and the file will not be split. This is the default. In the 'Compare groups' and 'Organize output by groups' options, data is split according to the groups (categories) of the categorical variable entered into the 'Groups Based on:' field. After selecting the desired split option, enter the desired categorical variable (Subjective Class Identification) into the 'Groups Based on:' field. Then, select 'Sort the file by grouping variables,' and click 'OK.'



The following image displays the Data View of the Data Editor split by 'class.' Note how the bottom right corner of the screen says 'Split by class.' Remember, all further analyses and graphs will reflect data split according to each class group (until the filter is turned off). For example, when a frequency table is generated, six individual frequency tables will be in the output, one for each class group (Lower, Working, Middle, Upper, DK, and NA).



**SPSS CODE FRAGMENT GENERATING ID NUMBERS WITHIN GROUP OR PANEL**

The code below shows how to create an id variable that takes on the values 1 to *n* within each group/panel (where *n* is the number of cases in a group).  The grouping variable is defined by the user.  The number of cases, *n*, need not be the same across all groups.  In the example dataset, **group** is the grouping variable.  The variables **v1** and **v2** are additional variables in the dataset and are not modified by the code.

**data list list /group v1 v2.**

**begin data**

**1 12 0**

**1 23 1**

**1 22 0**

**3 5 0**

**3 7 0**

**3 13 1**

**5 1 0**

**5 56 1**

**end data.**

**sort cases by group.**

**compute id = 1.**

**if group = lag(group) id = lag(id) + 1.**

**exe.**

**list group id v1 v2.**

group id v1 v2

1.00 1.00 12.00 .00

1.00 2.00 23.00 1.00

1.00 3.00 22.00 .00

3.00 1.00 5.00 .00

3.00 2.00 7.00 .00

3.00 3.00 13.00 1.00

5.00 1.00 1.00 .00

5.00 2.00 56.00 1.00

Number of cases read: 8 Number of cases listed: 8

### SORTING WITH SYNTAX

This syntax performs the same sort shown in the screencap above (sort by ascending class rank, then descending birth date).

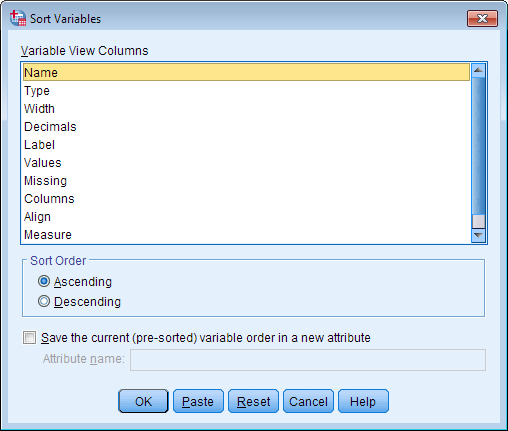
**SORT CASES BY Rank(A) bday(D).**

## Sort Variables

Sorting variables will rearrange the order of the variables (columns) in your data. Variables can be sorted on only one attribute at a time: Name, Type, Width, Decimals, Label, Values, Missing, Columns, Align, Measure, or a custom attribute. Variables can be sorted in ascending or descending order with respect to the selected attribute.

To sort variables, follow these steps:

1. Click **Data > Sort Variables**.
2. The Variable View Columns list includes the attributes of variables that may be used to sort variables, including: name, type, width, decimals, label, values, missing, columns, align, measure. Select an attribute by clicking it in the list, which will highlight your selection. Note that you can only select one variable attribute. In this example, “Name” has been selected, which means that variables will be sorted according to their names.

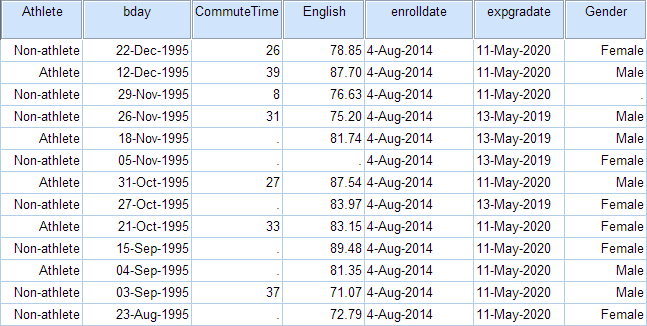


1. In the Sort Order area, variables can be sorted in “Ascending” or “Descending” order. Click the radio button that corresponds to your choice. In this example, variables will be sorted in ascending order.

You may also choose to save the current (pre-sorted) variable order in a new attribute by selecting the Attribute name check box and typing a name into the text field. Custom attributes are simply user-defined characteristics of variables that can be used to sort variables.

1. When you are finished, click **OK**.

Now your variables will be sorted according to the attribute you selected. In this example, the variables are sorted in ascending order according to their names (i.e., alphabetically).



### SYNTAX

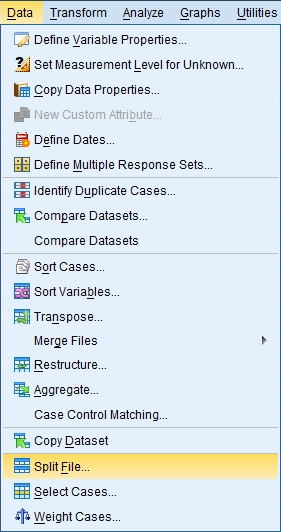
**SORT VARIABLES BY NAME (A).**

## Grouping or Splitting Data

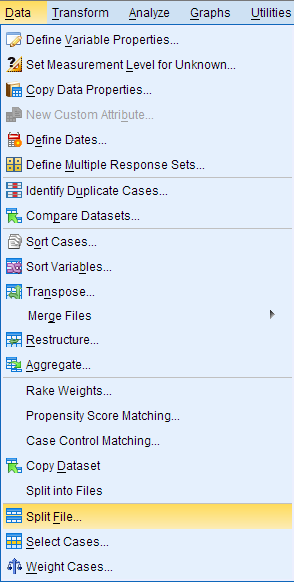
When analyzing data, it is sometimes useful to temporarily "group" or "split" your data in order to compare results across different subsets. This can be useful when you want to compare frequency distributions or descriptive statistics with respect to the categories of some variable (e.g., Gender) - especially if you want separate tables of results for each group.

To split your dataset, click **Data > Split File**.

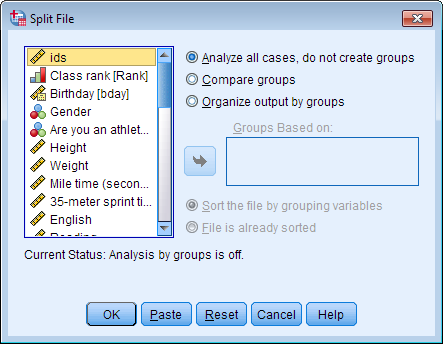
SPSS Version 21 Drop-Down Menu



SPSS Version 22 Drop-Down Menu



The Split File window will appear. By default, the dataset is not split according to any criteria; this is indicated by **Analyze all cases, do not create groups**.



You can choose one of two ways to split the data:

1. Compare groups
2. Organize output by groups

For both splitting methods, there are two considerations to be made:

* The splitting variable(s) should be nominal or ordinal categorical. SPSS will not stop you from using a continuous variable as a splitting variable, but it is a bad idea to try to attempt this; SPSS will see each unique numeric value as a distinct category.
* In order to split the file, SPSS requires that the data be sorted with respect to the splitting variable. By default, **Sort the file by grouping variables** is selected.

## Turning Off Split File

When you no longer want to split your analyses by group, you can turn Split File off through the same window you used to turn it on.

1. Click **Data > Split File.**
2. Click **Analyze all cases, do not create groups**.
3. Click **OK**.

You can now run all analyses normally again.

### SYNTAX

**SPLIT FILE OFF.**

## Example

What are the differences in the split file options?

The Compare and Organize options produce numerically identical results when the same grouping variable(s) are applied. This is true regardless of what statistical analysis is used. The difference between the two options is how the numeric results are presented.

* If **Compare groups**is used, then all of the results will be shown in a single table. The table will have sections showing the results for each group.
* If **Organize output by groups**is used, then each groups' results will be put into a separate table.

The choice of which splitting method to use is entirely about what format the user wants their results in. Do you want a single table with all results, or separate tables for each group's results? A good rule of thumb is to choose Compare Groups if you want to be able to directly compare the results of your groups, and to choose Organize Output by Groups if the information is from separate trials or samples (such as cohorts from different years).

### PROBLEM STATEMENT

Suppose that we want to get a summary of the differences in height between males and females in the sample data. Let's couple the Split File procedure with the Descriptives procedure to get summary statistics for the two groups. We'll use both Split File methods so that we can compare what their outputs look like.

### SPLITTING USING COMPARE GROUPS

If you choose to split your data using the **Compare groups** option and then run a statistical analysis in SPSS, your output will be displayed in a single table that organizes the results according to the grouping variable(s) you specified.

#### RUNNING THE PROCEDURE

To split the data in a way that will facilitate group comparisons:

1. Click **Data > Split File**.
2. Select the option **Compare groups**.
3. Double-click the variable Gender to move it to the **Groups Based on**field.
4. When you are finished, click **OK**.

After splitting the file, the only change you will see in the Data View is that data will be sorted in ascending order by the grouping variable(s) you selected.

Now let's view the aforementioned descriptive statistics for the variable Height with respect to Gender. Select **Analyze > Descriptive Statistics > Descriptives**. Double click on the Height variable, then click **OK**.

##### **Syntax**

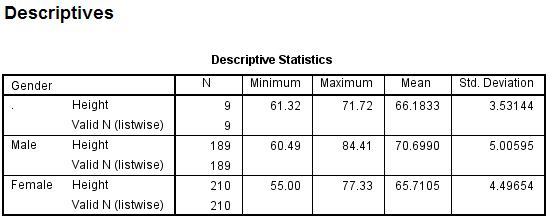
**SORT CASES BY Gender.**

**SPLIT FILE LAYERED BY Gender.**

**DESCRIPTIVES VARIABLES=Height**

**/STATISTICS=MEAN STDDEV MIN MAX.**

#### OUTPUT



This table gives us a breakdown of how many observations were in each group (N), and the minimum, maximum, average, and standard deviation of each group. The '.' group contains cases with missing gender values and nonmissing height values. At a glance, we can quickly take note that in this sample:

* The height of the tallest male was greater than the height of the tallest female.
* The male heights tended to have a slightly larger standard deviation (spread) than the female heights.
* On average, the males were taller than the females.
* The individuals with missing values for gender had a much smaller range of heights than did the males or females.

Note: This combination of Split File: Compare Groups with Descriptives is very similar to what you would get with the Compare Means procedure. The major difference is that Split File includes the missing values in the grouping/splitting variable, whereas Compare Means excludes missing values in the grouping variable.

### SPLITTING USING ORGANIZE OUTPUT BY GROUPS

If you choose to split your data using the **Organize output by groups** option and then run a statistical analysis in SPSS, your output will be broken into separate tables for each category of the grouping variable(s) specified.

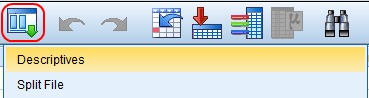
#### RUNNING THE PROCEDURE

To split the data in a way that separates the output for each group:

1. Click **Data > Split File**.
2. Select the option **Organize output by groups**.
3. Double-click the variable Gender to move it to the **Groups Based on**field.
4. When you are finished, click **OK**.

After splitting the file, the only change you will see in the Data View is that data will be sorted in ascending order by the grouping variable(s) you selected.

Now we will re-run the same descriptive statistics procedure that we ran before. You can go through the menu system again (**Analyze > Descriptive Statistics > Descriptives**), or you can click on the **Recall recently used dialogs** icon, which will bring up a list of recently used procedures:



##### **Syntax**

**SORT CASES BY Gender.**

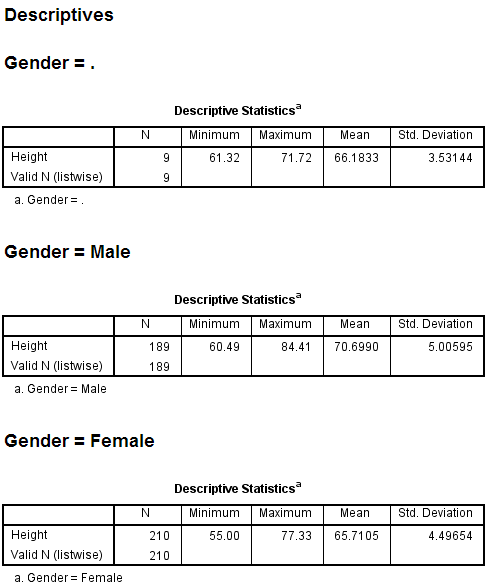
**SPLIT FILE SEPARATE BY Gender.**

**DESCRIPTIVES VARIABLES=Height**

**/STATISTICS=MEAN STDDEV MIN MAX.**

#### OUTPUT

After re-running the descriptive statistics, we see that the output is broken into three sections based on values of the Gender variable. The first section (“Gender = .”) reports the minimum, maximum, average, and standard deviation of Height for the students who had missing values for Gender. The second section reports those same statistics for the male students; the third section reports the statistics for the females.



# ARCHIVED: In SPSS, how do I drop cases with missing values?

The SELECT command with the SYSMIS() function can drop all missing cases from the current SPSS data set. Consider the following:

SELECT IF NOT (SYSMIS(amount)).

SAVE OUTFILE= 'newfile.sav'.

This example drops all cases whose value of the variable amount is missing, and then saves this data to an SPSS system file called newfile.sav.

If the data set has more than one coding for missing values, as is often the case for survey data, select all of the different codings for missing values with the AND operator:

SELECT IF NOT (SYSMIS(amount1)) AND NOT (SYSMIS(amount2)).

SAVE OUTFILE= 'newfile.sav'.

## SPSS IF Syntax Example 1

**\*1. Create test data.**  
data list free/gender score.  
begin data  
0 80 1 85 0 90 1 95 0 '' 1 105 0 110 1 115  
end data.  
  
**\*2. Replace missing value with 100.**  
if missing(score) score = 100.  
  
**\*3. Sort cases.**  
sort cases gender.

## SPSS IF Syntax Example 2

**\*1. Create score groups option 1.**  
if score lt 100 group\_a = 1.  
if score ge 100 group\_a = 2.  
exe.  
  
**\*2. Create score groups option 2.**  
recode score (100 thru hi = 2) (else = 1) into group\_b.  
exe.  
  
**\*3. Create score groups option 3.**  
compute group\_c = (score ge 100) + 1.  
exe.

## SPSS IF Syntax Example 3

**\*1. Gender-score groups option 1.**  
if score lt 100 and gender eq 0 group\_d = 1.  
if score ge 100 and gender eq 0 group\_d = 2.  
if score lt 100 and gender eq 1 group\_d = 3.  
if score ge 100 and gender eq 1 group\_d = 4.  
exe.  
  
**\*2. Gender-score groups option 2.**  
compute group\_e = 2 \* gender + (score ge 100) + 1.  
exe.